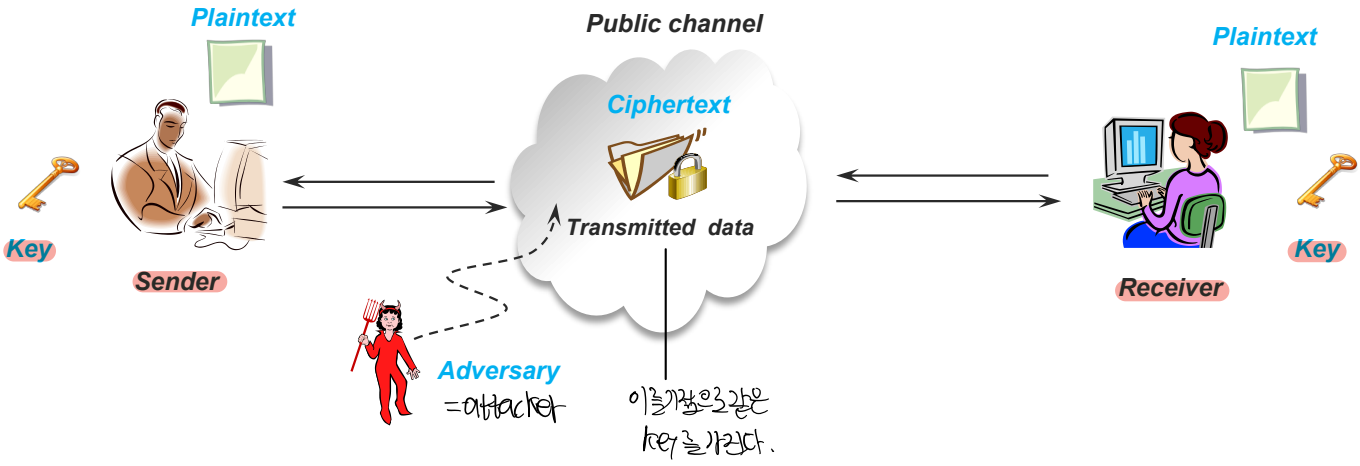
Classical Symmetric-key Ciphers

고전암호 (대칭키암호)

Jong Hwan Park

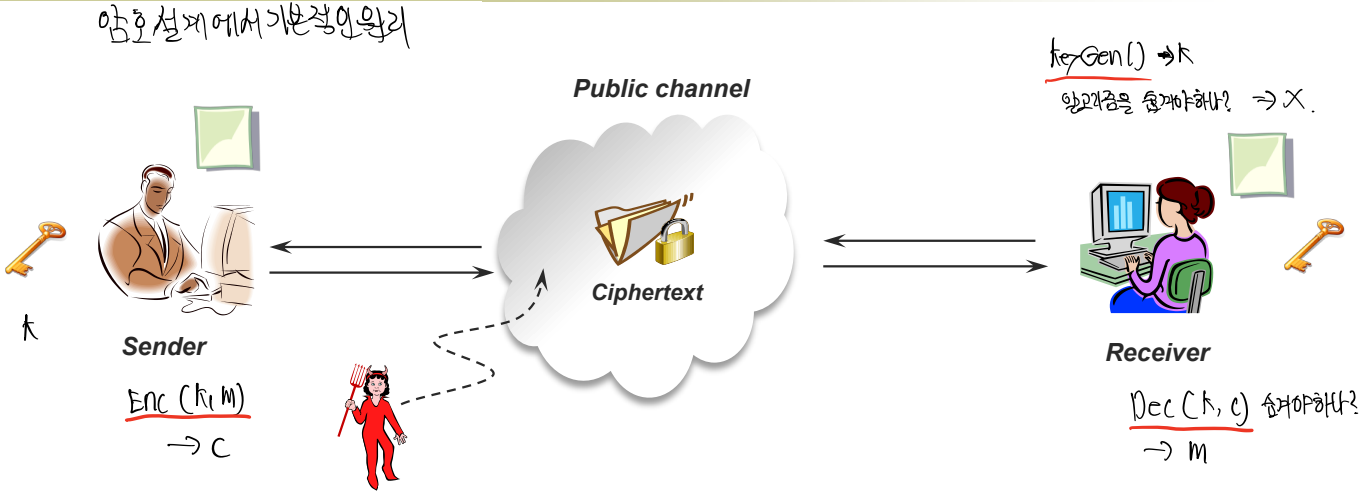
Symmetric-Key Encryption



- Definition of symmetric-key encryption
  - Key generation algorithm: Gen → a key $k \in KS$ (key space)
  - Encryption algorithm: Enc(k, m) → a ciphertext $C \in CS$ (ciphertext space)
  - Decryption algorithm: Dec(k, C) → a plaintext $m \in MS$ (message space)

Cf. Need to assume two parties initially share a key in a secure manner

Kerchhoffs's Principle (1)



- Clear to share the key secretly and keep it secret
- Should the Enc and Dec algorithms be kept secret, too?
- Kerchhoffs's opinion:

> "The cipher method must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience."

→ Demands that (1) security rely solely on the secrecy of the key, and (2) algorithms be made public

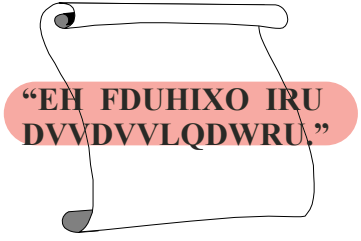
Kerchhoffs's Principle (2)

- (1) Why?
  - Much easier to share and securely store a short key than a (longer) program
  - Much easier to change the key than the (longer) algorithm in case the key is exposed (or refreshed) key를 주기적으로 refresh
  - Much easier to use the same algorithm/program, but with different keys, in case many pairs of people are involved
  - Reverse engineering of the code poses a serious threat

- (2) Why?
  - Published designs undergo public scrutiny
  - Better for security flaws to be revealed by "ethical cryptographers"
  - Enable the establishment of standards

Kerchhoffs's 원칙에 의해 암호 알고리즘은 OPEN, key 관리에 의해 안전성 확보 (공용 대칭키)

Caesar(shift) cipher (1)

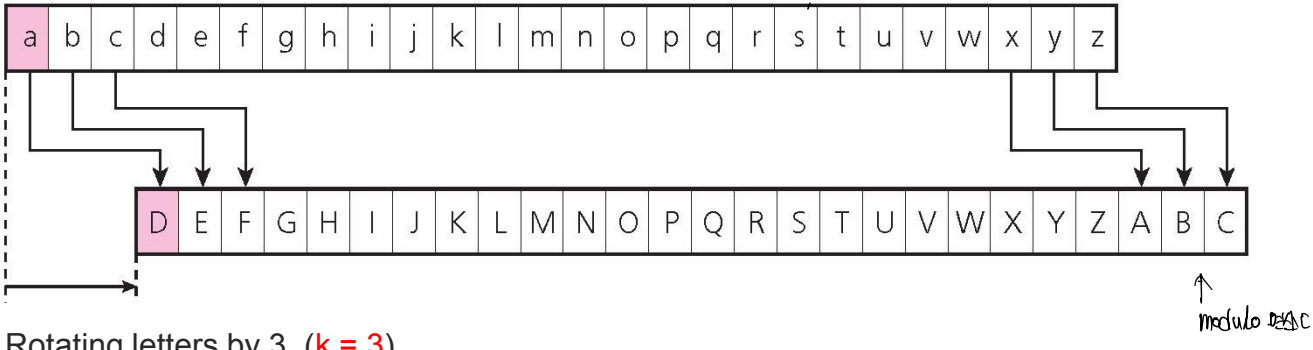


Ciphertext
(sent from Caesar)

Dec

Plaintext

Julius Caesar



Rotating letters by 3 (k = 3)

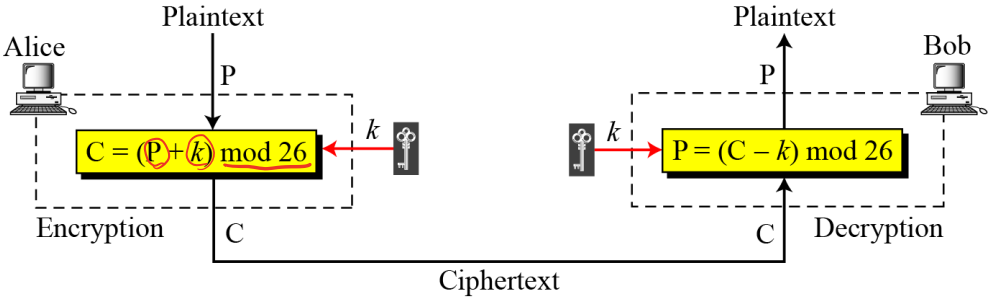
Caesar(shift) cipher (2)

- Plaintext and ciphertext in Z_{26} = {0, 1, 2, ..., 25}

Plaintext →	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext →	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

ex) 22 + 3 = 25 ≡ 0 (A)

- Mathematical description



- What is the problem?
  - a key ∈ {0, 1, …, 25} = Z_{26}; brute-force attack or exhaustive search !!
  - Need the sufficient key space principle

key space가 너무 작다.

고로 ⇒ key space를 충분히 크게 해야.

$||KS|| = 2^{128}$ → 128-bit security level

128-bit 이상 / 이하도

Mono-alphabetic Substitution Cipher

- Map each character to a different ciphertext character in an arbitrary manner

Plaintext →	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext →	N	O	A	T	R	B	E	C	F	U	X	D	Q	G	Y	L	K	H	V	I	J	M	P	Z	S	W

- Example of encryption

Plaintext → this message is easy to encrypt but hard to find the key

Ciphertext → ICFVQRVVNEFVRNVSIYRGAHSLIOJICNHTIYBFGTICRXRS

- Security
  - |KS| = 26! =26·25·24···2·1 (approximately 2^{88})
  - Easy to break this scheme even if |KS| is very large. Why?

Analysis of Mono-alphabetic Substitution Cipher (1)

- Average letter frequencies for English-language text

Letter	*Frequency*	*Letter*	*Frequency*	*Letter*	*Frequency*	*Letter*	*Frequency*
E	12.7	H	6.1	W	2.3	K	0.08
T	9.1	R	6.0	F	2.2	J	0.02
A	8.2	D	4.3	G	2.0	Q	0.01
O	7.5	L	4.0	Y	2.0	X	0.01
I	7.0	C	2.8	P	1.9	Z	0.01
N	6.7	U	2.8	B	1.5		
S	6.3	M	2.4	V	1.0		

영어 평균 출현에 대한 빈도수.

- Frequencies for diagrams and trigrams

Digram	TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS, OR, TI, IS, ET, IT, AR, TE, SE, HI, OF
Trigram	THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS, ETH, FOR, DTH

Analysis of Mono-alphabetic Substitution Cipher (2)

- Ciphertext

MEYLGVIWAMEYOPINYZGWYEGMZRUUYPZAIXILGVSIZZMPGKK
DWOMEPGROEIWGPCEIPAMDKKEYCIUYMGIFRWCEGLOPINYZHR
ZMPDNYWDWOGWITDWYSEDCEEIAFYYWMPIDWYAGTYPIKGLM
XFPIWCEHRZMMEYMEDWOMGQRYWCEUXMEDPZMQRGMEEYAP
ISDWOFICJILYSNICYZEYMGGJIPRWIWAIHRUNIWAHRZMUDZZYA
MEYFRWCEMRPWDWOPGRWAIOIDWSDMEIGWYMSGMEPYYEYH
RUNYARNFRMSDMEWGOPYIMYPZRCCYZZIOIDWIWAIOIDWEYMP
DYAILMYPMEYMWUNMDWOUGPZYKFRMIMKIZMEIAMGODTYD
MRNIWASIKJYAISIXSDMEEDZWGZYDWMEYIDPZIXDWODIUZRPY
MEYXIPYZGRPDMDZYIZXMGAYZNDZYSEIMXGRCIWWGMOYM

Analysis of Mono-alphabetic Substitution Cipher (3)

- Frequency of characters in the ciphertext

I	47	G	27	C	12	F	7	V	2
Y	47	Z	27	S	11	L	6	B	0
M	45	P	26	N	10	H	5		
W	35	R	22	U	10	J	3		
E	33	A	17	K	8	T	3		
D	30	O	16	X	8	Q	2		

Analysis of Mono-alphabetic Substitution Cipher (4)

- Substitution of the most frequent letter I or Y into e
- Assume that Y → e

T h

MEeLGVIWAMEeOPINeZGWeEGMZRUUePZAIXILGVSIZZMPGKKDW
OMEPGROEIWGPCEIPAMDKKEeCIUeMGIFRWCEGLOPINeZHRZMPD
NeWDWOGWITDWeSEDCEEIAFeeWMPIDWeAGTePIKGLMXFPIWCEH
RZMMEeMEDWOMGQReWCEUXMEDPZMQRGMEEeAPISDWOFICJIL
eSNICeZEeMGGJIPRWIWAIHRUNIWAHRZMUDZZeAMEeFRWCEMRP
WDWOPGRWAIOIDWSDMEIGWeMSGMEPeeEeHRUNeARNFRMSDME
WGOPeIMePZRCCeZZIOIDWIWAIOIDWEeMPDeAILMePMEeMWUNM
DWOUGPZeKFRMIMKIZMEIAMGODTeDMRNIWASIKJeAISIXSDMEE
DZWGZeDWMEeIDPZIXDWODIUZRPeMEeXIPeZGRPDMDZeIZXMGA
eZNDZeSEIMXGRCIWWGMOeM

Analysis of Mono-alphabetic Substitution Cipher (5)

theLGVIWAtheOPINeZGWehGtZRUUePZAIXILGVSIZZtPGKKDWOthPG
ROhIWGPChIPAtDKKheCIUetGIFRWChGLOPINeZHRZtPDNeWDWOG
WITDWeShDChhIAFeeWtPIDWeAGTePIKGLtXFPIWChHRZtthethDWOt
GQReWChUXthDPZtQRGthheAPISDWOFICJILeSNICeZhetGGJIPRWIW
AIHRUNIWAHRZtUDZZeAtheFRWChtRPWDWOPGRWAIOIDWSDthIG
WetSGthPeeheHRUNeARNFRtSDthWGOPeItePZRCCeZZIOIDWIWAIOID
WhetPDeAILtePthetWUNtDWOUGPZeKFRtItKIZthIAtGODTeDtRNIWASI
KJeAISIXSDthhDZWGZeDWtheIDPZIXDWODIUZRPetheXIPeZGRPDtDZ
eIZXtGAeZNDZeShItXGRCIWWGtOet

thPee

⇒ thhee

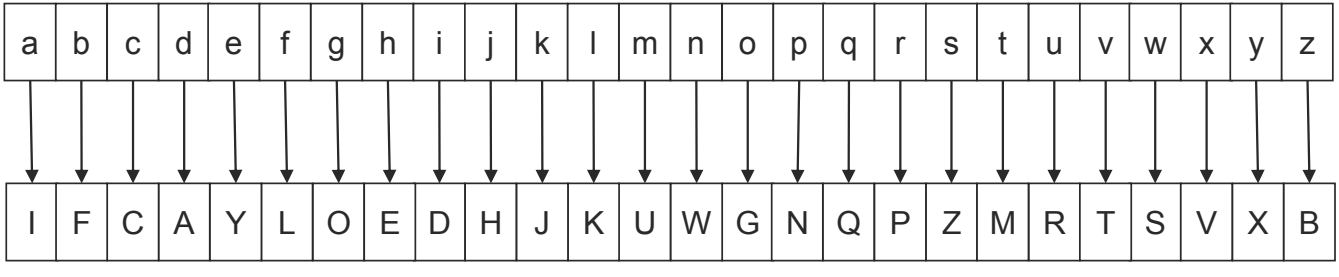
Analysis of Mono-alphabetic Substitution Cipher (6)

theLoVanAthegraNeZonehotZRUUerZAaXaLoVSaZZtroKKingthroRghanorc
harAtiKKhecaUetoaFRnchoLgraNeZHRZtriNeningonaTineShichhaAFeentrai
neAoTeraKoLtXFranchHRZtthethingtoQRenchUXthirZtQRothheAraSingFac
JaLeSNaceZhetooJarRnanAaHRUNanAHRZtUiZZeAtheFRnchtRrningroRnA
againSithaonetSothreeheHRUNeARNFRtSithnogreaterZRcceZZagainanAagai
nhetrieAaLterthetnUNtingUorZeKFRtatKaZthaAtogiTeitRNanASaKJeAaSaX
SithhiZnoZeintheairZaXingiaUZRretheXareZoRritiZeaZXtoAeZNiZeShatXo
Rcannotget

Shich
= which

Analysis of Mono-alphabetic Substitution Cipher (7)

thefoxandthegrapesonehotsummersdayafoxwasstroKKingthroughanorchard
tiKKhecametoaFunchofgrapesHustripeningonaTinewhichhadFeentrainedoT
eraKoftyFranchHustthethingtoquenchmythirstquothhedrawingFacJafewpac
eshetooJarunandaHumpandHustmissedtheFunchturningroundagainwithaone
twothreeheHumpedupFutwithnogreatersuccessagainandagainhetriedafterthe
tnmptingmorseKFutatKasthadtogiTeitupandwaKJedawaywithhisnoseinthea
irsayingiamsuretheyaresouritiseasytodespisewhatyoucannotget



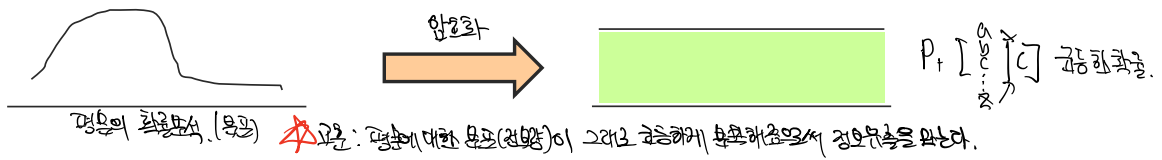
Analysis of Mono-alphabetic Substitution Cipher (8)

thefoxandthegrapesonehotsummersdayafoxwasstrollingthroughanorchardtilhecametoab
unchofgrapesjustripeningonavinewhichhadbeentrainedoveraloftybranchjustthetoquench
mythirstquothhedrawingbackafewpaceshetookarunandajumpandjustmissedthebunchtur
ningroundagainwithonetwothreehejumpedupbutwithnogreatersuccessagainandagainhetr
iedafterthetemptingmorselbutatlasthadtogiveitupandwalkedawaywithhisnoseintheairsay
ingiamsuretheyaresouritiseasytodespisewhatyoucannotget

"The Fox and the Grapes"

One hot summer's day, a Fox was strolling through an orchard till he came to a bunch f grapes just ripening on a vine which had been trained over a lofty branch. "Just the to quench my thirst, "quoth he. Drawing back a few paces, he took a run and a jump, and just missed the bunch. Turning round again with one , two, three, he jumped up, but with no greater success. Again and again he tried after the tempting morsel, but at last had to give it up, and walked away with his nose in the air, saying: "I am sure they are sour." It is easy to despise what you cannot get.

- Lesson:
  - Any information about plaintexts should not be revealed from ciphertexts



Vigenere (poly-alphabetic shift) Cipher

- To remove character frequencies in a ciphertext
- How it works:
  - Key = $(k_1, k_2, \ldots, k_d)$ (repeated as many times as needed)
  - $E_K(x) \equiv (x_1 + k_1, x_2 + k_2, \ldots, x_d + k_d)$
    $D_K(x) \equiv (c_1 - k_1, c_2 - k_2, \ldots, c_d - k_d)$

Plaintext →	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext →	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

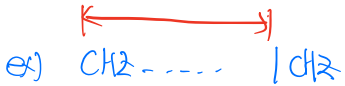
- Example of encryption: key = (PASCAL)=(15, 0, 18, 2, 0, 11)

Plaintext:	s	h	e	i	s	l	i	s	t	e	n	i	n	g
P's values:	18	07	04	08	18	11	08	18	19	04	13	08	13	06
Key stream:	*15*	*00*	*18*	*02*	*00*	*11*	*15*	*00*	*18*	*02*	*00*	*11*	*15*	*00*
C's values:	**07**	**07**	**22**	**10**	**18**	**22**	**23**	**18**	**11**	**6**	**13**	**19**	**02**	**06**
Ciphertext:	**H**	**H**	**W**	**K**	**S**	**W**	**X**	**S**	**L**	**G**	**N**	**T**	**C**	**G**

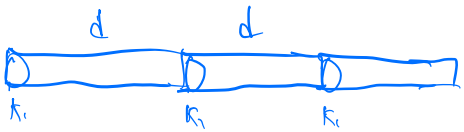
⇒ 동일문자 ⇒ 다른문자 대응 Mono-alphabetic cipher 의 공격해독.

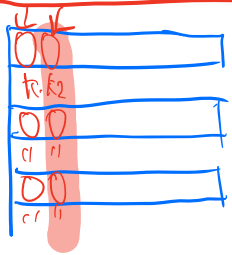
Analysis of Vigenere Cipher (1)

- Step 1
  - Find the length of the key
  - Kasiski test:
    - Search for repeated segments, of at least 3 characters, in the ciphertext
    - Find the distance d between two of segments
    - Using more segments, find $d_1, d_2, \ldots, d_n$
    - Key length could be one of the factors of $\gcd(d_1, d_2, \ldots, d_n)$



- Step 2 (if the key length m is found)
  - Divide the ciphertext into m pieces
  - Apply the frequency analysis to each piece used in mono-alphabetic substitution cipher





Analysis of Vigenere Cipher (2)

- Example of a ciphertext:

LIOMWGFEGGDVWGHHCQUCRHRWAGWIOWQLKGZETKKMEVLWPCZVGTH-
VTSGXQOVGCSVETQLTJSUMVWVEUVLXEWSLGFZMVVWLGYHCUSWXQH-
KVGSHEEVFLCFDGVSUMPHKIRZDMPHHBVWVWJWIXGFWLTSHGJOUEEHH-
VUCFVGOWICQLTJSUXGLW

- Kasiski test yields the results:

String	*First Index*	*Second Index*	*Difference*
JSU	68	168	100
SUM	69	117	48
VWV	72	132	60
MPH	119	127	8

- gcd(100, 48, 60, 8)=4 (which is a key length)

Analysis of Vigenere Cipher (3)

- Divide the ciphertext into 4 pieces:

```
C1: LWGWCRAOKTEPGTQCTJVUEGVGUQGECVPRPVJGTJEUGCJG
P1: jueuapymircneroarhtsthihytrahcieixsthcarrehe
C2: IGGGQHGWGKVCTSOSQSWVWFVYSHSVFSHZHWWFSOHCOQSL
P2: usssctsiswhofeaeceihcetesoecatnpntherhctecex
C3: OFDHURWQZKLZHGVVLUVLSZWHWKHFDUKDHVIWHUHFWLUW
P3: lcaerotnwhiwedssirsiirhketehretltiideatrairt
C4: MEVHCWILEMWVVXGETMEXLMLCXVELGMIMBWXLGEVVITX
P4: iardysehaisrrtcapiafpwtethecarhaesfterectpt
```

  - Key = (2, 14, 3, 4)=(c o d e)

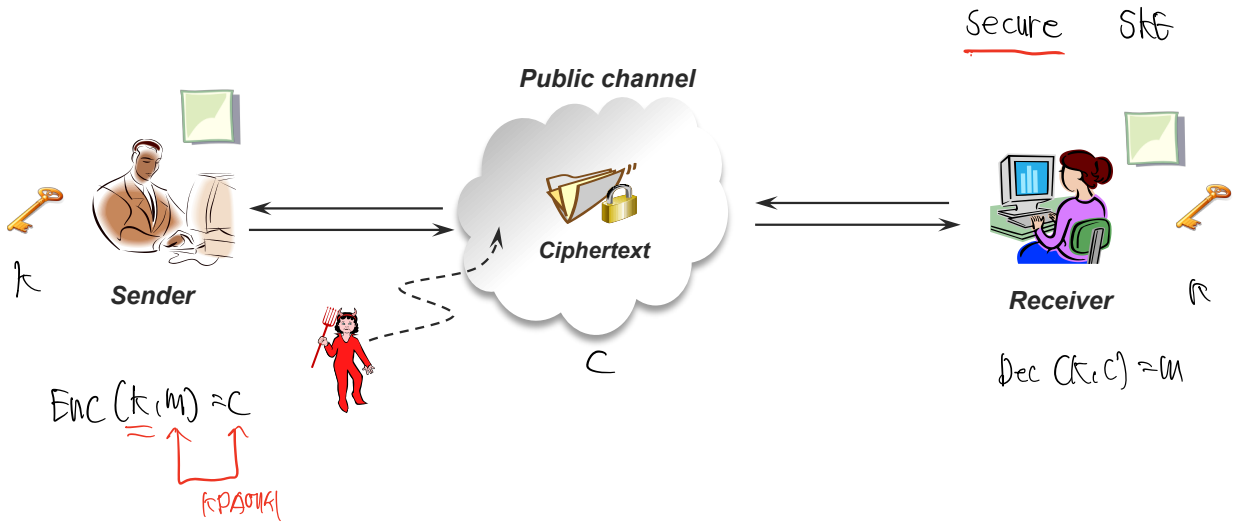
- The resulting plaintext makes sense:

Julius Caesar used a cryptosystem in his wars, which is now referred to as Caesar cipher. It is an additive cipher with the key set to three. Each character in the plaintext is shifted three characters to create ciphertext.

Important Lessons

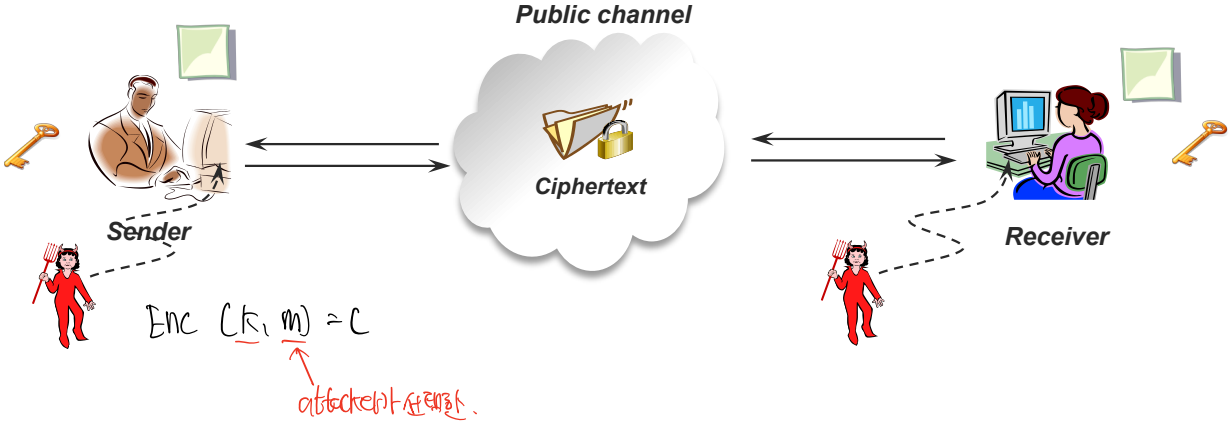
- Sufficient key space principle
- |KS| vs. |Ciphertext| required for analysis
  - |Key Space| : Caesar < Mono-alphabetic sub. < Vigenere
  - |Ciphertext| : Caesar < Mono-alphabetic sub. < Vigenere
- Designing secure ciphers is a hard task
  - All historical ciphers can be completely broken
  - Even far more complex schemes, such as the German *Enigma*

Attack Scenarios (1)



- Ciphertext-Only Attack (COA)
  - Just observe $\{C_i\}$) and aim to determine the underlying m corresponding to some target ciphertext
- Known-Plaintext Attack (KPA)
  - Learn $\{m_i, C_i\}$ under the same key and aim to determine the m corresponding to some target ciphertext

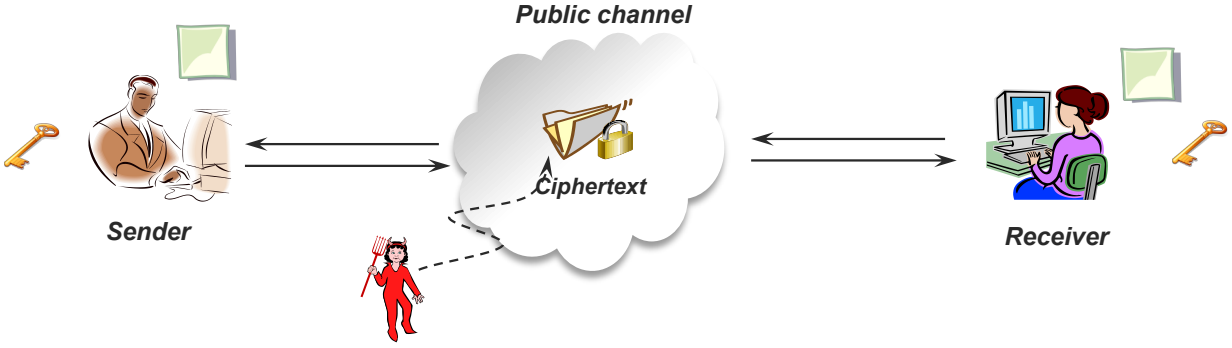
Attack Scenarios (2)



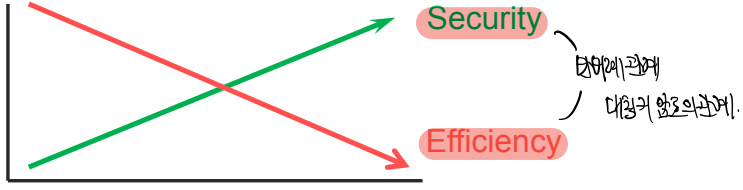
- Chosen-Plaintext Attack (CPA) 공격자가 원하는 평문을 선택해서 이를 개인에게 암호화를 한다.
  - Have the ability to obtain $\{C_i\}$ of plaintexts $\{m_i\}$ of its choice
  - Aim to determine the m corresponding to some target ciphertext
- Chosen-Ciphertext Attack (CCA) 1. 평문을 선택해 대응하는 암호문을 얻기 (CPA)
  2. 본인이 Dec 과정에서 복호화하는 암호문을 선택할 수 있다.
  - Have the ability to obtain $\{m_i\}$ of ciphertexts $\{C_i\}$ of its choice
  - Aim to determine the m corresponding to some target ciphertext

가장 강력.

Security & Efficiency

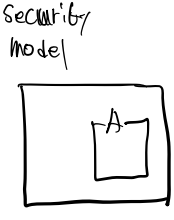


- Attack hardness: COA < KPA < CPA < CCA
- Not always the case that an encryption scheme secure against CCA should be used
  - May be less efficient than a scheme secure against *weaker* attacks
  - Tend to follow the fact that:



Two Approaches to Modern Cryptography

- How can one design cryptographic primitives?
  - Cryptanalysis-driven design
    - Problem → proposed solution → bug → revised solution → …
      → Implement → Bug → …
    - Symmetric-key encryption, MAC(Message Authentication Code), hash function, …
  - Provable-security approach
    - Problem → Definition → Scheme → Reduction → Implement → Done
    - Based on a complexity assumption
      - Infeasible to solve it in polynomial-time
      - {Diffie-Hellman, RSA, Discrete logarithm} problem …
    - Public-key encryption, digital signature, key agreement, authentication, ...



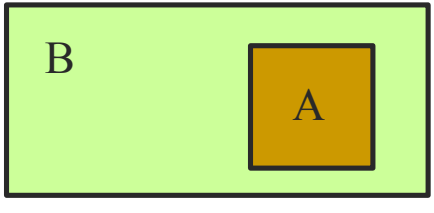
Basic Principle of Provable Security

- To solve any cryptographic problem
  - P1 - the formulation of a precise definition of security (i.e., attack model)
  - P2 - when the security relies on a complexity assumption, this assumption must be precisely stated
  - P3 - the security should be proven rigorously under the definition of (P1) and the assumption of (P2)

- Security proof in provable security

If the assumption X holds, a cryptographic scheme Y is secure (against a certain attack)

X → Y



Q & A